JEPIN
(Jurnal Edukasi dan Penelitian Informatika)
ISSN(e): 2548-9364 / ISSN(p) : 2460-0741

Vol. 7
No. 2
Agustus
2021

# Perbandingan Algoritma Klasifikasi terhadap Emosi *Tweet* Berbahasa Indonesia

Arif Bijaksana Putra Negara[#1], Hafiz Muhardi[#2], Fahmi Sajid[#3]

[#]*Program Studi Informatika, Fakultas Teknik, Universitas Tanjungpura*
*Jl.Prof. Dr.H.Hadari Nawawi, Pontianak, 78124*

[1]arifbpn@informatika.untan.ac.id
[2]hafizm@informatika.untan.ac.id
[3]fahmisajid@student.untan.ac.id

***Abstrak*— *Twitter* merupakan jejaring sosial dimana tempat orang-orang mengutarakan ekspresi dan emosi dirinya dalam bentuk tulisan. Mengidentifikasi emosi di *Twitter* tidak bisa menggunakan teknik text processing yang sederhana, karena kalimat pada *Twitter* yang singkat dan tata bahasa yang tidak teratur. Tujuan penelitian ini adalah untuk mengetahui algoritma yang mempunyai kinerja paling baik dalam menentukan klasifikasi emosi *tweet* pengguna *Twitter* Indonesia. Algoritma yang digunakan pada penelitian ini adalah Logistic Regression dan K-Nearest Neighbor. Dan juga pada penelitian ini melihat pengaruh TF-IDF dan Tuning Hyperparameter pada kedua algoritma. Hasil evaluasi menunjukkan bahwa performa model menggunakan algoritma logistic regression dengan *feature extraction* TF-IDF dan Tuning Hyperparameter memberikan perfoma model paling baik dengan nilai *accuracy* dan *f1-score* masing-masing sebesar 65% dan 66%. Model tersebut digunakan sebagai model prediksi *machine learning* untuk mengklasifikasikan dan mengelompokkan emosi-emosi pada aplikasi generik yang dibangun pada penelitian ini.**

***Kata kunci*— *Machine Learning, Text Mining, Emotion Classification, K-Nearest Neighbors, Logistic Regression***

## I. PENDAHULUAN

Media jejaring sosial merupakan tempat orang-orang mengekspresikan dirinya dalam bentuk tulisan, gambar, serta Video. *Twitter* Sebagai media sosial, *Twitter* mempunyai fungsi bertipe *micro-blogging* yaitu blog berukuran kecil dengan maksimal 140 karakter jumlah karakter dalam *tweet* atau post dalam *twitter* [1].

*Twitter* biasanya dimanfaatkan oleh instansi/lembaga atau perorangan untuk menyampaikan informasi dalam bentuk teks dan mendapatkan umpan balik dari apa yang disampaikan dalam informasi tersebut. Tidak hanya menyampaikan informasi saja, pengguna *twitter* juga dapat mengekspresikan emosinya [2]. Namun, emosi yang diungkapkan oleh masyarakat sangat beragam jenisnya. Strukutur kosa kata emosi dalam bahasa Indonesia telah didefinisikan oleh Shaver dan Murdaya [3]. Terdapat lima kelas emosi dalam bahasa Indonesia yaitu cinta (*love*), senang (*happiness*), marah (*anger*), kawatir/takut (*anxiety/fear*), dan sedih (*sadness*).

Mengidentifikasi emosi di *Twitter* tidak bisa menggunakan teknik text processing yang sederhana, karena kata pada *Twitter* yang singkat dan tata bahasa yang tidak teratur. Oleh karena itu dibutuhkan teknik *text mining* untuk mengekstrak kata" tidak terstruktur tersebut yang akan digunakan untuk membuat sebuah model prediksi *machine learning* yang dapat mengklasifikasikan dan mengelompokkan emosi-emosi tersebut, sehingga dapat mengurangi waktu dan biaya untuk mengidentifikasi emosi pada suatu kalimat. Algoritma klasifikasi terdiri dari beberapa cara, diantaranya menggunakan algoritma Naïve Bayes, Support Vector Machine, C4.5, Artificial Neural Network (ANN), *K-Nearest Neighbors* dan lainnya. Setiap algoritma mempunyai caranya tersendiri untuk mengklasifikasi, hasil kinerjanya juga bergantung pada kasus masalah, data yang digunakan, dan faktor lainnya.

Penelitian terkait klasifikasi emosi, khususnya teks Bahasa Indonesia telah dilakukan, diantaranya oleh: (a) Johanes et Al mengusulkan pendekatan dua tahap untuk deteksi emosi pada *tweet* Indonesia. Tahap yang pertama, *tweet* yang mengandung emosi dari sejumlah besar *tweet* mentah diekstraksi. Semua *tweet* yang diekstrak kemudian diklasifikasikan ke dalam lima kelas emosi yang telah ditentukan sebelumnya, yaitu cinta, kegembiraan, sedih, ketakutan, dan kemarahan menggunakan model komputasi berdasarkan pendekatan *machine learning* yang menggunakan berbagai bahasa, semantic dan ortografis [4]. (b) Saputri, dkk mengklasifikasikan emosi kedalam 5 kelas emosi (love, joy, anger, sadness, fear) dengan membandingkan algoritma klasifikasi *Logistic Regression*, SVM, dan Random Forest dan menggunakan kombinasi *feature extraction* [5]. Pada bahasa yang lain Alm et al. membahas masalah prediksi emosi berbasis teks dalam domain dongeng anak-anak menggunakan supervised *machine learning* [6].

Adapun penelitian terkait klasifikasi teks: Tomas & Viginjus menggunakan classifier Naive Bayes, Decision Tree, Support Vector Machine dan *Logistic Regression*

Submitted 23-07-2021; Revised 19-08-2021; Accepted 20-08-2021

untuk mengklasifikasi Text Review dari Amazon customer's dan Bag of word sebagai fitur ekstraksinya. Hasilnya kinerja classifier *Logistic Regression* lebih baik daripada classifier Naïve Bayes, Random Forest, Decision Tree, dan Support Vector Machines [7].

Muhammad Azam mengklasifikasikan scientific publications kedalam enam kelas (Medicine, Mathematic, Finance, Agricultural & Biological, Sciences, and Engineering) berdasarkan abstrak dan sitasi jurnal. Dataset yang digunakan sebanyak 10.000 abstrak. Algoritma klasifikasi yang digunakan yaitu Bayes dan *K-Nearest Neighbors* [8].

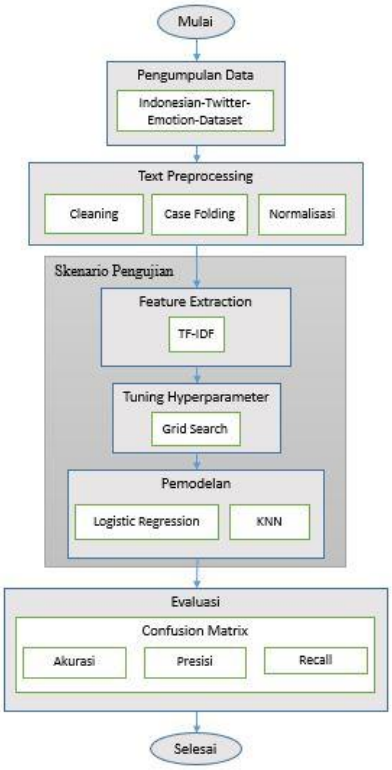
Teknik klasifikasi emosi yang digunakan pada penelitian ini adalah teknik klasifikasi *Logistic Regression* dan *K-Nearest Neighbors*. Kedua teknik klasifikasi ini akan dibandingkan dan kemudian teknik klasifikasi yang kinerjanya paling baik akan digunakan untuk membangun model dalam menentukan klasifikasi emosi pada aplikasi generik yang dibangun. Tidak hanya membandingkan teknik klasifikasi tetapi penelitian ini juga membandingkan hasil antara jika model menggunakan TF-IDF dan Tuning *Hyperparameter*.

## II. METODOLOGI

Terdapat lima tahap metode penelitian yang dilakukan pada penelitian ini yang dapat dilihat pada Gambar 1 untuk membangun model *machine learning*.

### A. Pengumpulan Data

Untuk membuat suatu model *machine learning* dibutuhkan suatu data pembelajaran. Data pembelajaran pada penelitian ini berupa teks. Data yang digunakan berasal dari penelitian Saputri, Mahendra, dan Adriani (2019) yang berjudul "*Emotion Classification on Indonesian Twitter Dataset*" [5]. Penelitian mereka bertujuan untuk membuat *Indonesian twitter dataset* untuk tugas klasifikasi dan untuk disebarluaskan. Data diambil menggunakan *Twitter* Streaming API sekitar 2 minggu, dimulai dari 1 Juni 2018 hingga 14 Juni 2018. Dataset ini memiliki 4403 Indonesian *Tweets* dimana sudah terlabel kedalam lima kelas emosi yaitu: "love", "anger", "sadness", "happy", dan "fear" (lihat Tabel 1). Contoh kalimat dapat dilihat pada Gambar 2.




Gambar. 1 Metode penelitian

```
label,tweet
"anger","Soal jln Jatibaru,polisi tdk bs GERTAK gubernur
.Emangny polisi tdk ikut pmbhasan? Jgn berpolitik. Pengaturan
wilayah,hak gubernur. Persoalan Tn Abang soal turun
temurun.Pelik.Perlu kesabaran. [USERNAME] [USERNAME]
[URL]""";"
"anger","Sesama cewe lho (kayaknya), harusnya bisa lebih
rasain lah yang harus sibuk jaga diri, rasain sakitnya haid,
dan paniknya pulang malem sendirian. Gimana orang asing?
Wajarlah banyak korban yang takut curhat, bukan dibela malah
dihujat.""";"
"happy","Kepingin gudeg mbarek Bu hj. Amad Foto dari google,
sengaja, biar teman-teman jg membayangkannya. Berbagi itu
indah.""";"
```

Gambar. 2 *Indonesian Twitter emotion dataset*

TABEL I
JUMLAH DATASET

| **Label** | **Indonesian-*Twitter*-Emotion-Dataset** |
|---|---|
| Anger | 1102 |
| Fear | 649 |
| Sadness | 998 |
| Love | 637 |
| Happy | 1017 |
| Total | 4403 |

### B. Text Preprocessing

Penelitian ini menggunakan *tools* Notebook Jupyter dengan menggunakan bahasa pemograman Python. Dataset yang digunakan adalah Indonesian-Twitter-Emotion-Dataset. Label adalah kelas emosi yang terdiri dari cinta, marah, sedih, senang, dan takut, sedangkan *tweet* merupakan kumpulan *tweet* dari pengguna *Twitter* berbahasa Indonesia. Data tersebut kemudian diinisialisasi

Korespondensi : Fahmi Sajid

ke dalam Notebook Jupyter untuk pra-pemrosesan teks. Hasil inisialisasi data dapat dilihat pada Gambar 2.

| | label | tweet |
|---|---|---|
| 0 | anger | Soal jln Jatibaru,polisi tdk bs GERTAK gubernu... |
| 1 | anger | Sesama cewe lho (kayaknya), harusnya bisa lebi... |
| 2 | happy | Kepingin gudeg mbarek Bu hj. Amad Foto dari go... |
| 3 | anger | Jln Jatibaru,bagian dari wilayah Tn Abang.Peng... |
| 4 | happy | Sharing pengalaman aja, kemarin jam 18.00 bata... |

Gambar. 2 *Indonesian Twitter emotion dataset*

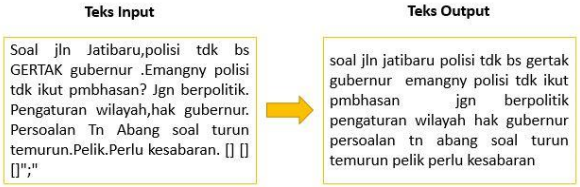
*Text Preprocessing* adalah tahap dimana data mentah ditransformasikan menjadi data yang tersetrukur sehingga dapat dibaca oleh anotator atau orang yang mengklasifikasi data. Data mentah yang di dapat dari pengguna dapat berisi data noisy yang tidak diperlukan. *Text Preprocessing* membersihkan data dengan menghapus URL, nama pengguna, stopword (kata yang kurang penting) dan semua data yang tidak relevan yang tidak mengekspresikan emosi apa pun untuk meningkatkan efisensi [9].

*Text Preprocessing* pada penelitian ini menggunakan bahasa pemrograman Python yang memanfaatkan library seperti Natural Language Toolkit (NLTK), Sastrawi, Pandas, dan lainnya. Adapun tahapan-tahapan *Text Preprocessing* yang dilakukan sebagai berikut:

*1) Cleaning:* Pada penelitian ini *Text Preprocessing* dimulai dengan proses *Cleaning*. *Cleaning* merupakan proses membersihkan text dari tag html, link, *script* dan komponen lainnya yang tidak memiliki kaitan informasi pada kalimat. Pada penelitian ini proses *Cleaning* yang dilakukan adalah menghapus term yang telah dilakukan *Text Preprocessing* sebelumnya yaitu [USERNAME], [URL], [SENSITIVE-NO] dan hashtag. Contoh *Cleaning* dapat dilihat pada Gambar 3.

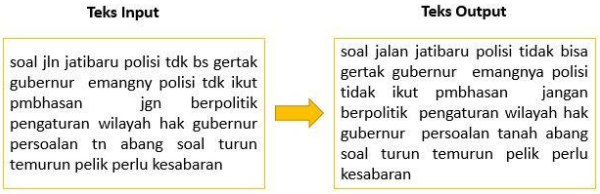

Gambar. 3 *Cleaning*

*2) Case Folding:* Tujuan dari *case folding* pada penelitian ini yaitu mengubah semua huruf pada kalimat menjadi huruf kecil. Contoh *case folding* dapat dilihat pada Gambar 4.




Gambar. 4 *Case folding*

*3) Normalisasi: Tweet* pada *Twitter* memiliki teks yang pendek dengan kata-kata tidak baku dan tata bahasa tidak terstruktur. Tujuan dari normalisasi adalah untuk mengubah kata yang tidak baku menjadi baku. Contoh Normalisasi dapat dilihat pada Gambar 5.




Gambar. 5 Normalisasi

## C. *Feature extraction*

*Feature extraction* adalah proses memilih serangkaian fitur dari beberapa cara mengurangi dimensi ruang fitur [10]. *Feature extraction* dapat meningkatkan *accuracy* dan mempersingkat waktu pembelajaran. Seleksi dari bagian kalimat dapat mencerminkan informasi tentang kata-kata konten, dan perhitungan bobot disebut ekstraksi fitur teks. *Feature extraction* yang digunakan dalam penelitian ini yaitu pembobotan TF-IDF.

Pembobotan TF-IDF terdiri dari dua proses yaitu atau *Term Frequency* (TF) dan Inverse *Document Frequency* (IDF). *Term Frequency* (TF) merupakan banyaknya kemunculan sebuah term atau kata pada suatu kalimat. Semakin banyak term atau kata dalam suatu kalimat, maka semakin besar nilai bobotnya [11]. Sedangkan *Inverse Document Frequency* (IDF) Menurut Riyanto merupakan banyaknya kemunculan term dalam kumpulan kalimat [12]. Berbeda dengan TF, pada IDF jika semakin banyak kemunculan term atau kata pada seluruh kalimat pada dataset, maka semakin kecil nilainya. jika semakin sedikit kemunculan term atau kata pada seluruh dokumen, maka semakin besar nilainya.

Pada penelitian ini untuk memulai proses TF-IDF hal pertama yang harus dilakukan yaitu membuat CountVectorizer untuk menghitung jumlah kata (Term Frequency) yang telah tersimpan pada variabel count_vect untuk diterapkan dengan menggunakan fit_transform pada variabel X_count (lihat Gambar 6). Kemudian untuk menghitung nilai IDF yaitu dengan memanggil TfidfTransformer yang telah tersimpan pada variabel tfidf_transformer untuk diterapkan dengan menggunakan fit_transform pada variabel X_tfidf.

```
count_vect = CountVectorizer()
X_count = count_vect.fit_transform(text)

tfidf_transformer = TfidfTransformer()
X_tfidf = tfidf_transformer.fit_transform(X_counts)
```

Gambar. 6 *Cleaning*

## D. *Tuning Hyperparameter*

Tuning *Hyperparameter* Tuning merupakan pemilihan *hyperparameter* yang optimal pada algoritma pembelajaran. Salah satu cara untuk memilih kombinasi *hyperparameter* yaitu dengan *Grid Search*. *Grid Search* akan mengkombinasikan hyperparamater-*hyperparameter* yang

Korespondensi : Fahmi Sajid

dimasukkan dan mencari kombinasi *hyperparameter* dengan nilai akurasi yang paling tinggi.

Tuning *Hyperparameter* pada penelitian ini dilakukan dengan menggunakan *Grid Search*. *Grid Search* pada python dilakukan dengan cara mengimport class GridSearchCV pada library sklearn. Pada penelitian ini *hyperparameter* yang diuji yaitu nilai C pada *Logistic Regression* dan nilai K pada K-Nearest Neighbor.

*E. Pemodelan*

Terdapat dua Pendekatan yang dipilih untuk melakukan klasifikasi pada penelitian ini, yaitu *Logistic Regression* dan K-Nearest Neighbor.

*1) Logistic Regression: Logistic Regression* merupakan model regresi untuk menganalisis hubungan antara suatu variabel dependent dan satu atau lebih variabel independent. Variabel dependent terdiri dari dua kategori yaitu "ya (sukses)" dan "tidak (gagal)" dan dinotasikan 1="sukses" dan 0="gagal" [13],[14],[15]. Pemodelan *Logistic Regression* pada penelitian ini memanfaatkan fungsi LogisticRegression dari library Sklearn. Berikut Bentuk model *Logistic Regression*:

$$Log(\frac{P}{1-P}) = \beta 0 + \beta\, 1X1 + \beta\, 2X2 + \ldots + \beta KXK \quad (1)$$

Sebelumnya *hyperparameter* yang sesuai pada *Logistic Regression* telah dicari menggunakan Grid Search. *Hyperparamaters* yang diuji yaitu nilai C.

*2) K*-Nearest *Neighbor:* K-Nearest Neighbor adalah metode klasifikasi objek terhadap data pelatihan dengan pemilihan kelas objek berdasarkan jarak terdekat dari data pelatihannya. Berdasarkan atribut dan sample dari data training algoritma ini dapat mengklasifikasikan objek baru. KNN adalah metode klasifikasi dengan cara menghitung nilai K (*Neighbor*). K merupakan titik neighbor objek, nilai K akan dibandingkan untuk menentukan klasifikasi objek tersebut [16],[17]. Pemodelan *K-Nearest Neighbors* pada penelitian ini dilakukan dengan menggunakan bahasa pemrograman Python yang memanfaatkan fungsi KNeighborsClassifier dari *library* Sklearn. Tahapan untuk menjalani proses KNN sebagai berikut (Alm, Roth, and Sproat 2005):

- Menentukan jumlah pada tetangga k.
- Menghitung jarak objek dengan masing-masing data kelompok. Perhitungan jarak menggunakan rumus euclidian distance [18] yang ditunjukan pada persamaan 4. Dimana D adalah jarak, x dan y adalah data latih dan data uji.

  $$D(x,y) = \sqrt{\sum_{k=1}^{n} (xi - yi)^2} \quad (2)$$
- Lalu didapatkan hasil pengklasifikasian.

Sebelumnya *hyperparameter* yang sesuai K-Nearest Neighbor telah dicari menggunakan Grid Search. Hyperparamaters yang diuji yaitu nilai K.

*3) Skenario Pengujian:* Terdapat empat skenario pengujian pada masing-masing algoritma *Logistic Regression* dan K-Nearest Neighbor (lihat Tabel 2). Skenario pengujian pertama dilakukan menggunakan *feature extraction* TF dan algoritma klasifikasi dengan default *hyperparameter*. Skenario pengujian kedua dilakukan menggunakan *feature extraction* TF-IDF dan algoritma klasifikasi dengan default *hyperparameter*. Skenario pengujian ketiga dilakukan menggunakan *feature extraction* TF dan algoritma klasifikasi dengan Tuning *Hyperparameter*. Skenario pengujian keempat dilakukan menggunakan *feature extraction* TF-IDF dan algoritma klasifikasi dengan tuning *hyperparameter*.

TABEL II
JUMLAH DATASET

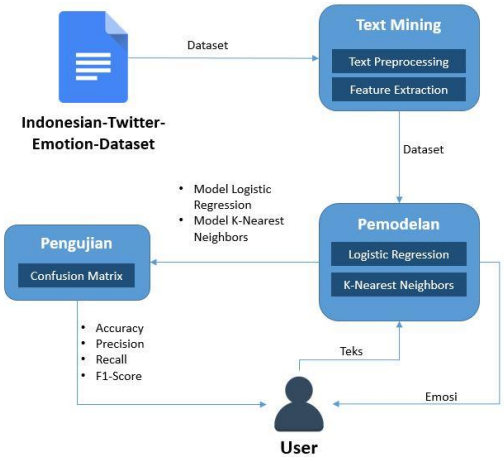
| Skenario Pengujian | *Tuning Hyperparameter* | *Feature extraction IDF* |
|---|---|---|
| Skenario pengujian pertama | × | × |
| Skenario pengujian kedua | × | ✓ |
| Skenario pengujian ketiga | ✓ | × |
| Skenario pengujian kedua | ✓ | ✓ |

*F. Evaluasi*

Pada penelitian ini *10-Fold Cross validation* digunakan dalam membagi data set secara bergantian menjadi data test dan data training untuk mendapatkan nilai prediksi data test. Kemudian hasil prediksi dari mesin dibandingkan dengan data label untuk di evaluasi menggunakan *Confusion Matrix*. Untuk menjelaskan keterhubungan antar bagian dalam model klasifikasi maka dapat dilihat dari arsitektur sistem yang dibangun. Arsitektur sistem digunakan untuk menjelaskan rencana atau pemetaan dari proses-proses yang akan dilakukan dalam membangun emotion mining machine.

Gambar 7 merupakan arsitektur sistem pada penelitian ini. Langkah pertama, Data yang telah dikumpulkan dari penelitian yang telah dipublikasikan oleh putri dkk [5] dilakukan *Text Mining* untuk membersihkan dan mengekstrak kalimat-kalimat yang ada pada dataset. Kemudian dari data tersebut dijadikan bahan pembelajaran untuk membuat model *Machine Learning*. Hasil pemodelan kemudian dilakukan pengujian dengan *Confusion Matrix* untuk mendapatkan nilai *Accuracy*, *Precision*, *Recall*, dan *F1-score* model *machine learning*. Selain itu user dapat melakukan percobaan dengan memasukkan sebuah teks/kalimat dan model akan melakukan prediksi emosi terhadap teks/kalimat masukkan.

Korespondensi : Fahmi Sajid




Gambar. 7 Arsitektur sistem

## III. HASIL DAN PEMBAHASAN

Tahap-tahap yang dilakukan untuk membuat model *machine learning* mulai dari pengumpulan data, *preprocessing*, *feature extraction*, hingga pemodelan telah dilakukan. Terdapat delapan model dibuat pada penelitian ini. Data uji pada setiap model dibagi menggunakan *10-Fold Cross Validation*. Hasil Prediksi data uji dibandingkan dengan label untuk dievaluasi menggunakan *Confusion Matrix*.

### A. Hasil Implementasi Text Preprocessing

Hasil sample dataset dari keseluruhan *Text Preprocessing* ditampilkan dalam bentuk tabel yang dapat dilihat pada Tabel 3.

TABEL III
HASIL TEXT PREPROCESSING

| ***Text*** | ***Text Preprocessing*** |
|---|---|
| Soal jln Jatibaru,polisi tdk bs GERTAK gubernur .Emangny polisi tdk ikut pmbhasan? Jgn berpolitik. Pengaturan wilayah,hak gubernur. Persoalan Tn Abang soal turun temurun.Pelik.Perlu kesabaran. [USERNAME] [USERNAME] [URL]";" | Tanpa Preprocessing |
| Soal jln Jatibaru,polisi tdk bs GERTAK gubernur .Emangny polisi tdk ikut pmbhasan? Jgn berpolitik. Pengaturan wilayah,hak gubernur. Persoalan Tn Abang soal turun temurun.Pelik.Perlu kesabaran. [] [] [];" | *Cleaning* |
| soal jln jatibaru polisi tdk bs gertak gubernur emangny polisi tdk ikut pmbhasan jgn berpolitik pengaturan wilayah hak gubernur persoalan tn abang soal turun temurun pelik perlu kesabaran | Case Folding |
| soal jalan jatibaru polisi tidak bisa gertak gubernur emangnya polisi tidak ikut pmbhasan jangan berpolitik pengaturan wilayah hak gubernur persoalan tanah abang soal turun temurun pelik perlu kesabaran | Normalisasi |

### B. Hasil Implementasi Text Preprocessing

Berikut adalah Gambar 8 yang menampilkan hasil dari *Feature extraction*.

```
(0, 17565)    0.1961799014492943
(0, 17042)    0.16370942989995824
(0, 16655)    0.10180128382076828
(0, 16189)    0.22919374954525334
(0, 15960)    0.16722835983563678
(0, 15402)    0.29199571554679665
(0, 13274)    0.33445671967127355
(0, 13235)    0.22919374954525334
(0, 12999)    0.19266097151361578
(0, 12936)    0.14000549218022634
(0, 12657)    0.2185085969474145
(0, 12448)    0.22919374954525334
(0, 7683)     0.17986646713992016
(0, 6571)     0.2185085969474145
(0, 6533)     0.10205964670324753
(0, 6494)     0.12196338391260503
```

Gambar. 8 Hasil TF-IDF

Keterangan:
1. Document Index
2. Word Index
3. TF-IDF Score

### C. Hasil Implementasi Hyperparameter Tuning

Model yang tidak dan menggunakan TF-IDF menghasilkan nilai dari Tuning *Hyperparameter* yang berbeda. Hasil Tuning *Hyperparameter* dapat dilihat Pada Tabel 4.

TABEL IV
HASIL *HYPERPARAMETER* TUNING

| **Algoritma Klasifikasi** | **Tanpa TF-IDF** | **Menggunakan TF-IDF** |
|---|---|---|
| *Logistic Regression* | C = 0.08858667904100823 | C = 1.623776739188721 |
| *K-Nearest Neighbor* | K = 44 | K = 43 |

### D. Hasil Evaluasi Logistic Regression

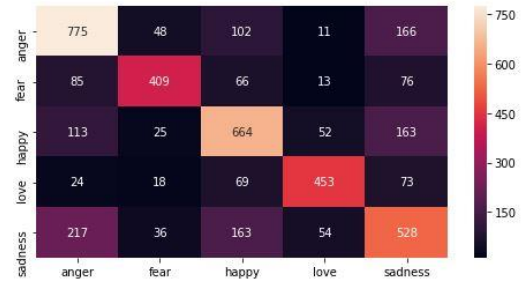
Berikut adalah Tabel 5 yang menampilkan hasil evaluasi dari keempat model *Logistic Regression*. Confusion Matrix model pertama hingga keempat dapat dilihat pada Gambar 9-12.

TABELV
HASIL EVALUASI MODEL *LOGISTIC REGRESSION*

| **Performa Model** | **Model Pertama** | **Model Kedua** | **Model Ketiga** | **Model Keempat** |
|---|---|---|---|---|
| *Accuracy* | 64% | 64% | 64% | 65% |
| *Precison* | 67% | 69% | 67% | 69% |
| *Recall* | 65% | 63% | 64% | 64% |
| *F1-score* | 65% | 65% | 65% | 66% |


Gambar. 9 Confusion matrix model pertama *logistic regression*


Gambar. 10 Confusion matrix model kedua *logistic regression*




Gambar. 11 Confusion matrix model ketiga *logistic regression*


Gambar. 12 Confusion matrix model keempat *logistic regression*

## *E. Hasil Evaluasi Model K-Nearest Neighbors*

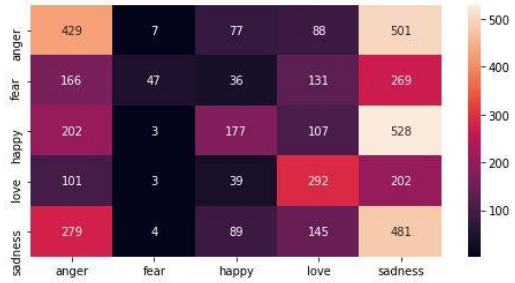
Berikut adalah Tabel 6 yang menampilkan hasil evaluasi *Logistic Regression*. Confusion Matrix model pertama hingga keempat dapat dilihat pada Gambar 13, 14, 15, 16.

TABEL VI
HASIL EVALUASI MODEL *K-NEAREST NEIGHBORS*

| **Performa Model** | **Model Pertama** | **Model Kedua** | **Model Ketiga** | **Model Keempat** |
|---|---|---|---|---|
| *Accuracy* | 33% | 49% | 34% | 55% |
| **Performa Model** | **Model Pertama** | **Model Kedua** | **Model Ketiga** | **Model Keempat** |
| Precison | 39% | 48% | 42% | 56% |
| *Recall* | 32% | 50% | 30% | 57% |
| *F1-score* | 31% | 48% | 29% | 55% |


Gambar. 13 Confusion matrix model pertama K-Nearest Neighbor


Gambar. 14 Confusion matrix model kedua K-Nearest Neighbor




Gambar. 15 Confusion matrix model ketiga K-Nearest Neighbor


Gambar. 16 Confusion matrix model keempat K-Nearest Neighbor

Korespondensi : Fahmi Sajid

*F. Analisis Logistic Regression*

Hasil evaluasi pada *Logistic Regression* (lihat Tabel 5) menunjukkan bahwa model keempat memperoleh hasil performa yang paling baik dengan nilai *accuracy* dan *f1-score* 65% dan 66%. Model keempat yaitu model yang menggunakan TF-IDF dan *Hyperparameter* C=1.623776739188721. Confusion Matrix model keempat pada *Logistic Regression* dapat dilihat pada Gambar 12.

Selain itu, Hasil evaluasi *Logistic Regression* pada model kedua menunjukkan pengaruh TF-IDF. Hasil dari model kedua menunjukkan bahwa dengan TF-IDF nilai *precision* mengalami kenaikan sebesar 2,985% namun pada nilai *recall* mengalami penurunan sebesar 3,077%.

Hasil evaluasi *Logistic Regression* pada model ketiga menunjukkan pengaruh Tuning *Hyperparameter*. Hasil dari model ketiga menunjukkan bahwa dengan Tuning *Hyperparameter* nilai *precision* mengalami penurunan 1,538%. Dan hasil dari model keempat menunjukkan bahwa dengan TF-IDF dan Tuning *Hyperparameter* nilai *accuracy*, *precision*, dan *F1-score* masing-masing mengalami kenaikan sebesar 1,5625%; 2,985% dan 1,538% namun nilai *recall* mengalami penurunan sebesar 1,538%.

Di sisi lain, Confusion Matrix *Logistic Regression* pada keempat model menunjukkan kesalahan prediksi yang paling banyak terjadi yaitu kalimat pada kelas Sadness yang terprediksi sebagai kelas Anger dan Happy, akibatnya nilai *recall* menjadi rendah. Kemudian kesalahan yang paling banyak terjadi juga kalimat pada kelas Anger dan Happy yang terprediksi kelas Sadness, akibatnya nilai *precision* menjadi rendah.

*G. Analisis K-Nearest Neighbors*

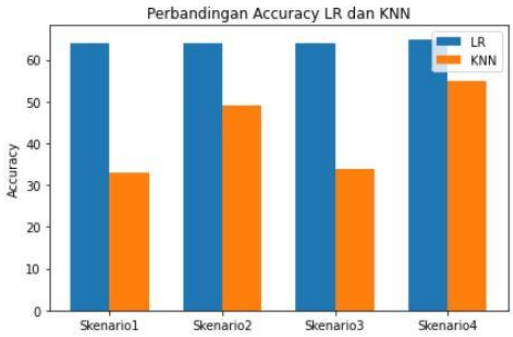
Hasil evaluasi pada *K-Nearest Neighbors* (lihat Tabel 4.4) menunjukkan bahwa model keempat memperoleh hasil performa yang paling baik dengan nilai *accuracy* dan *f1-score* 55% dan 55%. Model keempat yaitu model yang menggunakan TF-IDF dan *Hyperparameter* K=43. Confusion Matrix Skenario keempat pada *K-Nearest Neighbors* dapat dilihat pada Gambar 4.9.

Confusion Matrix pada model keempat menunjukkan bahwa rendahnya nilai akurasi disebabkan oleh banyaknya kalimat yang tidak terprediksi tidak tepat sesuai dengan kelasnya. Kesalahan prediksi yang paling banyak terjadi yaitu pada kelas sadness yang terprediksi sebagai kelas anger dan happy, akibatnya nilai *recall* menjadi rendah. Kemudian kesalahan prediksi yang paling banyak terjadi juga pada kelas happy dan anger yang terprediksi sebagai kelas anger, akibatnya nilai *precision* menjadi rendah.

Selain itu, hasil dari model kedua menunjukkan bahwa dengan TF-IDF nilai *accuracy*, *precision*, *recall*, dan *f1-score* masing-masing mengalami kenaikan sebesar 48,5%; 23,08%; 56,25%; dan 54,84%. Dan hasil dari model ketiga menunjukkan bahwa dengan Tuning *Hyperparameter* nilai *accuracy* dan *precision* masing-masing mengalami kenaikan sebesar 3,03% dan 7,69% namun nilai *recall* dan *f1-score* masing-masing mengalami penurunan sebesar 6,25% dan 6,45%. Dan Hasil dari Skenario 4 menunjukkan bahwa dengan TFIDF dan Tuning *Hyperparameter* nilai *accuracy*, *precision*, *recall*, dan *f1-score* masing-masing mengalami kenaikan sebesar 66,7%; 43,6%; 78,125; dan 77,42%.

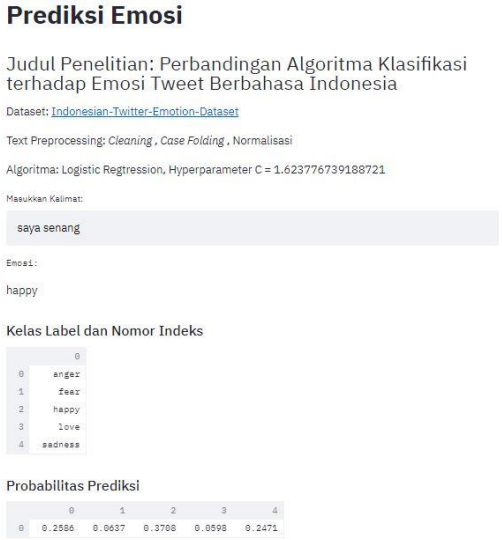
*H. Perbandingan Hasil Evaluasi Logistic Regression dan K-Nearest Neighbors*

Hasil evaluasi *Logistic Regression* (lihat Tabel 5) dan *K-Nearest Neighbors* (lihat Tabel 6) menunjukkan bahwa *Logistic Regression* pada model keempat menghasilkan performa terbaik dengan nilai *accuracy* dan *f1-score* masin-masing sebesar 65% dan 66%. Sedangkan algoritma *K-Nearest Neighbors* memiliki nilai *Accuracy* dan *f1-score* masing-masing sebesar 55% dan 55%.




Gambar. 17 Perbandingan *Accuracy* LR dan KNN

Hasil evaluasi *Logistic Regression* pada keempat model memiliki hasil performa yang lebih baik dibandingkan menggunakan algoritma *K-Nearest Neighbors* (lihat Gambar 17). Namun penggunaan TF-IDF atau Tuning *Hyperparameter*s tidak memiliki pengaruh terhadap nilai *accuracy Logistic Regression*. Dan pada skenario 4 hanya meningkatkan *accuracy* sebesar 1,5625%. Disisi lain penggunaan TF-IDF dan Tuning *Hyperparameter*s pada *K-Nearest Neighbors* dapat meningkatkan nilai *accuracy* dan *f1-score* masing-masing hingga 66,7% dan 77,42%.




Gambar. 18 Tampilan aplikasi

*I. Implementasi aplikasi generik*

Aplikasi generik yang dibangun berdasarkan model dengan hasil evaluasi paling baik yaitu *Logistic Regression* dengan *Hyperparameter* C=1.623776739188721 dan *feature extraction* TF-IDF. Aplikasi generik yang dibangun berbasis web dengan fitur prediksi dengan mengisi teks kedalam kolom input. Kemudian setelah diinput teks hasil output aplikasi yaitu prediksi emosi dari teks yang diinputkan dan probabilitas prediksi tersebut. Tampilan dari Aplikasi generik dapat dilihat pada Gambar 18. Aplikasi generik dapat di akses pada link https://predictemotion.herokuapp.com/.

## IV. KESIMPULAN

Berdasarkan hasil penelitian yang telah dilakukan maka dapat disimpulkan sebagai berikut.

1. Performa model *logistic regression* jika menggunakan TF-IDF dan Tuning *Hyperparameter* memberikan perfoma pengujian paling baik dengan nilai *accuracy* dan *f1-score* masing-masing sebesar 65% dan 66%. Sedangkan hasil pengujian pada algoritma K-Nearest Neighbor memberikan performa *accuracy* dan *f1-score* sebesar 55%.
2. Penggunaan *feature extraction* Tf-Idf lebih berpengaruh terhadap algoritma *K-Nearest Neighbors* yang dapat meningkatkan nilai *accuracy* dan *f1-score* masing-masing sebesar 48,4% dan 54,84% dibandingkan dengan *Logistic regression* yang hanya dapat meningkatkan nilai *precision* sebesar 2,985%.
3. Penggunaan Tuning *Hyperparameter* lebih berpengaruh terhadap kedua algoritma apabila menggunakan *feature extraction* TF-IDF. Penggunaan TF-IDF dan *Tuning Hyperparameter* dapat meningkatkan nilai *accuracy* dan *f1-score* pada *Logistic Regression* masing-masing sebesar 1,5625% dan 1,538% dan pada *K-Nearest Neighbor* masing-masing sebesar 66,7% dan 77,42%.

## REFERENSI


[1] Phuvipadawat, S. and Murata, T. Breaking news detection and tracking in Twitter. Proceedings - 2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology - Workshops, WI-IAT:120–123.

[2] Hirat, R. 2015. A Survey On Emotion Detection Techniques using Text in Blogposts. International Bulletin of Mathematical Research. 2(1):180-187

[3] P. R. Shaver, U. Murdaya, and R. C. Fraley, "Structure of the Indonesian Emotion Lexicon," Asian Journal of Social Psychology, vol. 4, no. 3, pp. 201–224, 2001.

[4] The. J. E., A. F. Wicaksono, and M. Adriani. 2015. A two-stage emotion detection on indonesian tweets. In Advanced Computer Science and Information Systems (ICACSIS). Pages 143–146.

[5] Saputri, Mei Silviana, Rahmad Mahendra, and Mirna Adriani. 2019. "Emotion Classification on Indonesian Twitter Dataset." Proceedings of the 2018 International Conference on Asian Language Processing, IALP 2018 (January 2019):90–95.

[6] Alm, Cecilia Ovesdotter, Dan Roth, and Richard Sproat. 2005. "Emotions from Text: Machine Learning for Text-Based Emotion Prediction." HLT/EMNLP 2005 - Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference (October):579–86.

[7] Pranckevičius, Tomas, and Virginijus Marcinkevičius. 2017. "Comparison of Naive Bayes, Random Forest, Decision Tree, Support Vector Machines, and Logistic Regression Classifiers for Text Reviews Classification." Baltic Journal of Modern Computing 5(2):221–32.

[8] Azam, Muhammad, Tanvir Ahmed, Fahad Sabah, and Muhammad Iftikhar Hussain. 2018. "*Feature extraction* Based Text Classification Using K-Nearest Neighbor Algorithm." IJCSNS International Journal of Computer Science and Network Security 18(12):95–101.

[9] Pinto, Joylin Priya, and Vijaya Murari. 2008. "Real Time Sentiment Analysis of Political Twitter Data Using Machine Learning Approach." International Research Journal of Engineering and Technology 4124.

[10] D Trier, AK Jain, dan T Taxt. 1996. *Feature extraction* methods for character recognition—a survey. Pattern Recogn. 29(4), 641–662.

[11] Muslim, AA. 2016. Rancang Bangun Aplikasi Berbasis Web untuk Analisis Sentimen pada Mikroblog Twitter dengan Metode Naive Bayes. Electronic Theses Universitas Islam Negeri Maulana Malik Ibrahim.

[12] Riyanto, A. 2016. Text Summarization dengan Metode K-Means Pada Artikel Berita Berbahasa Indonesia. Digital Library UNIKOM.

[13] Rachimawan, Achmad Fachrudin. 2016. "ADS Filtering Menggunakan Jaringansyaraf Tiruan Perceptron, Naïve Bayes Classifier Dan Regresi Logistik." 5(1):98.

[14] Saputra, Rizal Amegia. 2014. "Komparasi Algoritma Klasifikasi Data Mining Untuk Memprediksi Penyakit Tuberculosis ( Tb ): Studi Kasus Puskesmas Karawang." Seminar Nasional Inovasi Dan Tren (SNIT) (April):1–8.

[15] Intansari, Ida Ayu Sevita, Santi Wulan Purnami, and Sri Pingit Wulandari. 2012. "Klasifikasi Pasien Hasil Pap Smear Test Sebagai Pendeteksi Awal Upaya Penanganan Dini Pada Penyakit Kanker Serviks Di RS. 'X' Surabaya Dengan Metode Bagging Logistic Regression." Jurnal Sains Dan Seni ITS 1(1):D277–82.

[16] Rahmadianto, Rizky, Edy Mulyanto, and T. Sutojo. 2019. "Implementasi Pengolahan Citra Dan Klasifikasi K-Nearest Neighbor Untuk Mendeteksi Kualitas Telur Ayam." Jurnal VOI (Voice Of Informatics) 8(1):45–54.

[17] Abdurrahman, Muhammad Hanif, Efri Suhartono, and Eka Wulandari. 2019. "Deteksi Kualitas Kemurnian Susu Sapi Melalui Pengolahan Citra Digital Menggunakan Metode Scale Invariant Feature Transform ( Sift ) Dan Klasifikasi K-Nearest Neighbor ( Knn ) Quality Detection of Cow ' S Milk Purity Using Digital Image Processing Method O." 6(2):3845–52.

[18] Sreemathy. J., dan Balamurungan. P. S., 2012. An efficient text classification using KNN and naïve bayesian. International Journal on Computer Science and Engineering (IJCSE), vol. 4, no. 3, pp. 392-396.


Korespondensi : Fahmi Sajid